**Assignment 2.1**

Create a mongo db collection representing a team of players for a sport. This could be made up data or data based on a real team for a sport of your choosing.

For example, maybe I choose the GreenBay packers football team. I want to create a collection of the players on the team (or perhaps a subset of the player if the team is very large, you can assume a limit of 15 players for this assignment). I would keep track of a few major statistics from last season.

Maybe I keep track of rushing yards, touchdowns thrown, sacks, made and missed field goals, and catches made.

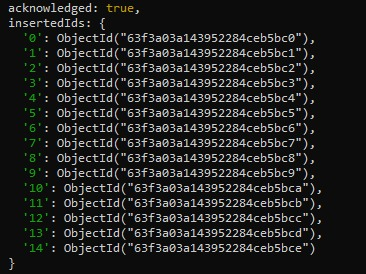
Ans: Here I created the Table with the name of **team** please find the screenshot below as it shows **{ok: 1}** which means Table created Successfully.



Next, I inserted **15** members data, please find the below link to find the sample data I have inserted

[**https://justpaste.it/b4607**](https://justpaste.it/b4607)

Once I inserted the sample data It will Shows the output as **acknowledged: true** following with different **insertIds** please refer below screenshot.



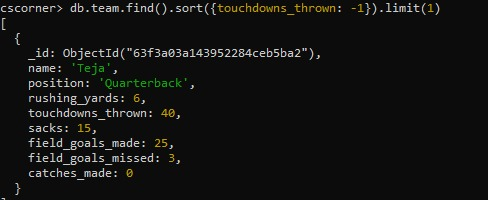
Once you have created your collection, create and demonstrate 5 queries that would be appropriate for your collection.

For example, maybe I want to query my collection for:

1. The Player with the most touchdown passes.

**Query: db.team.find().sort({touchdowns\_thrown: -1}).limit(1)**

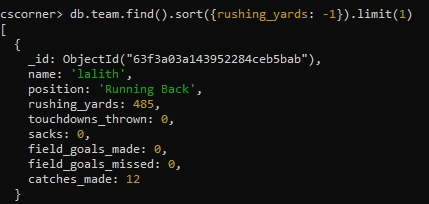
**Ans: Output for the above query will create new ObjectId along with the Output for the above sample data.**



1. The Player with the most rushing yards?

**Query: db.team.find().sort({rushing\_yards: -1}).limit(1)**

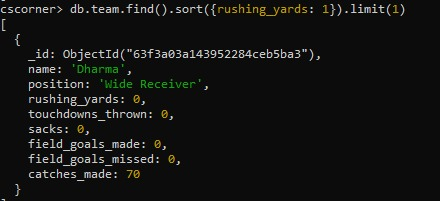
**Ans: Output for the above query will create new ObjectId along with the Output for the above sample data.**



1. The Player with the least rushing yards.

**Query: db.team.find().sort({rushing\_yards: 1}).limit(1)**

**Ans: Output for the above query will create new ObjectId along with the Output for the above sample data.**



1. A list of players sorted from most to fewest field goals made.

**Query: db.team.find().sort({field\_goals\_made: -1})**

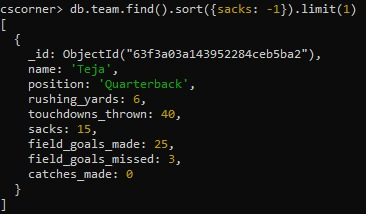
Ans: Its displays all the Team Players in Descending Order. So, I’m unable to paste the entire output here please refer the below Link to see the output from the external link which I pasted.

[**https://justpaste.it/bqnx4**](https://justpaste.it/bqnx4)

1. The Player with the most number of sacks.

**Query: db.team.find().sort({sacks: -1}).limit(1)**

**Ans: Output for the above query will create new ObjectId along with the Output for the above sample data.**



While these examples come nowhere close to using the power of MongoDB, they do show the common types of queries a RESTful api might demand of a MongoDB collection.

**Ans: Yes, that's correct! The examples I provided are just a few common types of queries that a RESTful API might require to retrieve data from a MongoDB collection.**

Submit screenshot of evidence that you have completed the work along with the self-evaluation document found on Blackboard.

**Ans: Yes, I submitted screenshots of all outputs please Refer above.**

Answer the following questions and paste it into the top of the submission box for each homework assignment for the class.

1. How long did you spend on this assignment?

**Ans: I have spent nearly 1 hour.**

1. Based on your effort, what letter grade would you say you earned? Based on your solution, what letter grade would you say you earned?

**Ans: I spend few hours only to do this Assignment where all the questions are basic level only. So, I thought that the letter grade I would that I have earned “B” because still I have to learn more.**

Provide a summary of what doesn’t work in your solution, along with an explanation of how you attempted to solve the problem and where you feel you struggled.

**Ans: in starting while I writing the queries, I don’t know how to use inbuilt functions and their syntax, so I referred to the online sources and MongoDB Documentation. I didn’t face any struggles while I solving the problems because I loved to Learn this.**